GENERATION OF ROTATIONAL SWEEP SHADOWS FOR POLYHEDRONS

NAGW-1333

by

Henry L. Welch and Robert B. Kelley

Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering
Troy, New York 12180-3590

August, 1990

CIRSSE REPORT #60

Submitted for publication.

Generation of Rotational Sweep Shadows for Polyhedrons

HENRY L. WELCH and ROBERT B. KELLEY

Center for Intelligent Robotic Systems for Space Exploration
Electrical, Computer and System Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Abstract -- The determination of sweep shadows is important when analyzing the potential interference effects of obstacles in a robotic environment. This paper presents multiple techniques for generating the rotational sweep shadows of polyhedral objects through the use of a sweep plane. Various implementation difficulties are discussed with proposed solutions presented.

Introduction

The order in which an assembly is performed can drastically affect the overall time it takes to perform that assembly. A good ordering can reduce the number of assembly errors and manufacturing difficulties. For example, some alternate assembly sequences may require less fixturing or fewer changes of tools and grippers than others. It may be possible, by assembling different parts at different times, to develop sequences which have mating trajectories with fewer and more distant obstacles to avoid. Such a choice could thereby result in simpler and more reliable assembly operations.

It is the goal of this paper to answer some of the questions posed by the calculation of geometric feasibility. Geometric feasibility determines whether or not two subassemblies can be properly mated along a collision free trajectory. Related to this calculation is the determination of the gripper envelope, or the volume available for the gripper, during an assembly operation. The goal of these types of calculations is to determine the potential interference effects of obstacles in the environment.

Background

A significant body of research has been compiled in recent years which addresses the question of assembly sequence planning. In most cases the basic approach is to use the geometric relationships between parts and the idea of geometric feasibility to generate a list of all the possible feasible assembly sequences. Differences between the various approaches involve the method by which assembly sequences are represented and the degree and type of operator interaction required by the algorithm.

Two of the earliest attempts, by Ko and Lee [Ko] and Fox and Kempf [Fox], use a precedence graph to represent the relationship between the various assembly tasks and require the operator to supply all the geometric feasibility information. Further work by De Fazio and Whitney [De Fazio 87, De Fazio 88] uses directed graphs of assembly states and provides a consistent set of operator questions for determining geometric feasibility. Later, Homem de Mello and Sanderson [Homem 86, Homem 88] propose the use of AND/OR graphs of subassemblies to represent the assembly sequences and provide an algorithm for analyzing geometric feasibility.

A common element missing from all these approaches is the ability to rank the various assembly sequences so as to be able to determine the sequence most likely to be successful. Among the factors which may be used to rank sequences and specific operations in the sequence are the size and shape of the gripper envelope and the subassembly stability (in the presence of friction and gravity).

Figure 1 shows a proposed blueprint for the architecture of an assembly sequence planner. Many of the modules depicted exist, in one form or another, as research efforts throughout the literature [Welch]. The notable exception is the module for performing gripper envelope analysis. In this paper the issues related to determining the bounds on the gripper envelope are addressed.

The Gripper Envelope

Humans and robots are both similar in that each need an envelope or volume in which to perform actions. In most cases the volume required for an action is centered about the object being acted upon and changes as the operator moves the object past obstacles in the environment. There are two techniques which can be used to deal with the problem. The first requires full geometric data on the robot or agent performing the assembly. An algorithm for detecting collisions, for example, is one proposed by Mirolo and Pagello [Mirolo]. The second technique performs the motion

without the presence of the agent and generates the volume not occupied by obstacles for later analysis. This second technique is the one being considered here because it provides a non-robot-specific solution.

Initially the calculation of the gripper envelope closely resembles the volume sweeping problem, as described by Wang and Wang [Wang], in which the obstacles in the environment sweep out volumes which cannot be occupied by either the part being mated or the mechanism moving the part. However, by limiting the types of motions that an object may follow, the problem can be simplified.

Problem Simplifications

As is commonly the case in both robotics and other fields a problem is first partially solved by making restrictions to the shapes of geometric entities. In the field of robotics this usually involves restrictions to the types of paths possible in mating trajectories and to the shapes of objects being represented. Another common assumption in robotics is that the geometric relationship between that object and the hand/fingers of the agent remains fixed from the time an object is grasped until the time it is released. This eliminates many types of uncertainty that are difficult to model.

The most common limitation on the shapes of geometric entities is to limit objects to the set of convex polyhedrons as demonstrated by Mirolo and Pagello [Mirolo]. A less restrictive class of objects would be those possessing planar faces. This is a common technique in surface modeling and yields accurate approximations of most objects [see Blinn, Foley, and Turner]. This is the class of objects used throughout the rest of this paper.

From basic mechanics, it is known that all forms of motion can be broken up into two distinct components. The first component is tangential to the direction of motion and the second component is normal to it. Generalizing the concept to volume sweeping divides the volume being swept into a cross-sectional area perpendicular to the direction of sweeping and a distance along the direction of sweeping.

The cross-sectional area of the gripper with respect to the mating trajectory is constant due to the fixed relationship between the object and the robot agent's hand and the limitation of the mating trajectories to paths with uniform tangential components, . Examples of trajectory paths which fit this criterion are straight line segments, simple rotations, helical paths, and constant radius curves. Thus, by looking at the projected sweep shadow cast by obstacles in the direction of the path tangent, an estimate of the cross-section available for a robot's hand can be made.

These shadows can be generated by sweeping a plane along the direction of the motion and marking the cross-sectional area of the obstacles encountered by the plane. For the case of straight line segments, this is a projection in a cartesian coordinate system and is trivial to compute with the exception of clipping and closure at the ends of the path segments. The case of simple rotations is not as obvious and provides the body of the paper below.

The Rotational Sweep Shadow Problem

The goal of the rotational sweep shadow problem is to compute the areas in the radius-height plane which are swept by a full plane sweeping through an angle in the specified rotational reference frame given the arbitrary rotational reference frame, the angle to rotate through, and the environment of obstacles. More specifically, given an arbitrary axis of rotation, center of the rotation space, a reference point defining the starting location of the sweep plane, and an angle to rotate the plane through, sweep the plane through the rotation and record all the areas on the plane that are *swept out* by obstacles in the environment.

The rotation of a full plane is necessary since it is unknown which half of the sweep space is important relative to the reference point. For example, consider a hand drill with a T shaped handle. While it is sufficient to represent its orientation by specifying the location of one side of the handle, both sides of the handle may encounter obstacles when it rotates.

Upon further consideration, this problem reduces to the projection of data (through the angular coordinate) in an arbitrary cylindrical coordinate system with the added consideration of clipping at the initial and final location of the sweep plane. There are also some other complications brought about since the plane being swept is a full plane and not the half-plane usually associated with a cylindrical coordinate system. The two most obvious complications are the inclusion of data points containing both positive and negative radii (something not allowed in a strict cylindrical coordinate system) and, when the rotational angle is greater than π, points on an obstacle may appear in both the positive and negative sides of the sweep plane. These are some of the features which make this problem interesting.

The Basic Solution

Having defined the rotational sweep shadow problem, it is now possible to explore solutions to the problem. Since the sweeping operation reduces to projection through the angle of rotation in a cylindrical coordinate system, it is advantageous to use cylindrical coordinates in the solution of the problem. Since clipping and closure are also considerations, algorithms to accomplish these operations are needed. Figure 2 shows a simplified description of the algorithm to be used.

The next few sections describe in detail the various calculations necessary to implement the algorithm.

Defining the Cylindrical Coordinate System

The initial step in solving the rotational sweep shadow problem is to develop a representational formalism that simply and compactly defines the relevant data. Figure 3 shows how an arbitrary point in the environment can be represented in terms of a cylindrical coordinate system defined by the rotation.

One problem with the formalism of Figure 3 is that it does not specify the angular component of the point P. To do this, a cartesian coordinate system must be built around the cylindrical system so that a reference direction for the angular component can be defined. The first obvious choice of axis is to use the axis of rotation as the new cartesian z-axis. Specifying the new cartesian x- and y-axes requires more thought. Since a reference point representing the starting location of the rotation plane has already been specified it would be advantageous to use this to define the new cartesian x-axis. In general the direction from the center of the rotation to the reference point is not orthogonal to the already chosen z-axis. The solution to this is to use the Gram-Schmidt orthonormalization technique to specify the x-axis orthogonal to the existing z-axis [Hoffman, p. 280]. The y-axis then follows naturally in the right hand sense as the vector cross product between the z-axis and the x-axis.

Converting Data to the Sweep Reference System

Now that the sweep reference system is defined it is necessary to convert all the environment data to that coordinate system. Letting T represent a coordinate transformation matrix from the global coordinate system to the sweep reference system using the x-, y-, and z-axes defined in the previous section. A matrix notation can be used to convert the coordinate P, as defined in Figure 3,

to the r, θ, and h of the sweep reference system by first finding the scalar components, $[a\ b\ c]$, of the position vector P in the sweep reference system as shown below [Selby, p. 369].

$$[P - P_r] = [a\ b\ c][T] \rightarrow [a\ b\ c] = [P - P_r]\ [T]^{-1}$$

$$r = \sqrt{a^2 + b^2}$$

$$\theta = tan^{-1}(\frac{b}{a})$$

$$h = c$$

Clipping Edges Against the Rotational Sweep Wedge

In a rectangular system the clipping of edges against a fixed planar boundary is straightforward and yields relatively few difficulties [Hearn, pp. 128-134]. Extending the idea of clipping to two parallel planar faces involves clipping against each of the planar faces individually. It would appear that the same basic idea can be used to clip an edge against the two constant θ-planes which bound the rotational sweep wedge; however, this is not entirely true.

If the calculation of the rotational sweep shadow involved only the rotation of a half-plane then clipping against the two constant θ sides of the wedge would sufficiently clip the end or ends of each edge. However, a consideration of both positive and negative wedges reveals that edges swept by the negative radius half of the sweep plane must be clipped to the opposite sides of the constant θ wedge boundaries. This requires clipping of each edge against four boundaries to solve the problem. It is also possible to clip each edge against the two positive radius wedge boundaries by rotating the edge data through π radians and then reclipping against the same two boundaries. This second technique is the one that is used here and is essentially the same as performing the clipping operations for two half plane rotational sweeps.

Another approach to the clipping problem involves clipping against both constant θ wedge boundaries simultaneously. This is not as straightforward as the rectangular case because of the wrap-around nature of angular data. (Angles greater than twice π are not possible.) In the rectangular case, an indication would be given as to whether an edge's endpoints are beyond one of the two boundaries and which of the two boundaries it is beyond. In the case of cylindrical data, it is not readily apparent whether the endpoint is beyond the $\theta = 0$ or $\theta = \theta_{max}$ wedge boundary. This effect can be counteracted by also considering the midpoint of each edge. Figure 4 shows all the possible arrangements of endpoints and midpoints for the case when the first endpoint as represented by $p1$ has a θ value greater than the second endpoint which is represented by $p3$ Solution of the problem when the role of the endpoints is reversed follows by symmetry.

If either of these techniques is used, a method is still needed to calculate the intersection point with the wedge boundary. By parameterizing the edge with endpoints $p1$ and $p2$ the following vector formulation is obtained:

$$[p_1 + m(p_2 - p_1)] = [a\ b\ c];\ 0 \leq m \leq 1$$

For the case of finding the intersection point with the $\theta = 0$ wedge boundary the coordinate b becomes zero. By introducing the vector d= [0 1 0] and by taking the vector dot product of d with both sides of the above equation the following condition results [Selby, p. 540]:

$$p_1 \cdot d + m(p_2 \cdot d - p_1 \cdot d) = 0$$

Solving for m yields:

$$m = \frac{p_1 \cdot d}{p_1 \cdot d - p_2 \cdot d}$$

Solving for the intersection point with the $\theta = \theta_{max}$ wedge boundary is straightforward using the same basic approach and a simple trick. If the coordinates of the edge are rotated about the z-axis by negative θ_{max} radians, then the y-coordinate of the intersection point (b above) becomes zero. To do this, the vector d needs to be changed to $d = [-sin(\theta_{max})\ cos(\theta_{max})\ 0]$ and the above calculations repeated [Hearn, pp. 108-109].

Since it is already known that the edge under study intersects the wedge boundary (due to previous calculations), the above solution holds unless p_1 and p_2 are identical.

Closing Faces Against the Rotational Sweep Wedge

In the previous section, two different methods are presented which demonstrate how an edge can be clipped against the constant θ boundaries of the sweep wedge. In the first method, the edges are clipped against each planar boundary separately. Closure of each face along the clipping plane is routine provided that the edges bounding the face are processed in an orderly fashion [Hearn, pp. 134-138].

For the second method where clipping against both boundaries is done simultaneously, the results of clipping and the necessary closure are not as clear cut. Figure 5 shows the four types of closure possible for the positive radius wedge. In Figure 5a and 5b the method of closure is identical to that used when clipping a face against a single plane. Figure 5c shows a polygon which is closed against both faces and Figure 5d depicts a polygon that leaves from one of the wedge boundaries and enters via the other. In this case, the resultant closure requires the addition of two edges instead of the more typical one.

Unlike the normal closure case, the clipping algorithm does not supply all the information necessary to construct the two new edges of Figure 5d. The data missing is the value of the z or height, h, of the face where the radius is zero. According to calculus, a plane can be defined by a normal vector to the plane and a point on the plane by the function:

$$(p - p_0) \cdot n = 0$$

where p_0 is the given point on the plane and n is the normal to the plane. Knowing that the radius must be zero, and hence both x and y must be zero, reduces the formula to:

$$[\langle 0\ 0\ h \rangle - \langle x_0\ y_0\ z_0 \rangle] \cdot [n_x\ n_y\ n_z] = 0$$

and

$$h = x_0 \frac{n_x}{n_z} + y_0 \frac{n_y}{n_z} + z_0$$

the clipping and closure algorithms guarantee that the face crosses the radius-equals-zero point, the value of n_z must be nonzero.

Discovered Inadequacies of the Current Solution

At first glance it would appear that all the necessary elements are now in place to generate the complete rotational sweep shadow for a group of objects and an arbitrary rotational reference. Unfortunately, this is not the case. There are two main problems which need to be addressed. The first is caused by values of the rotation angle greater than π radians. The reason that this is a problem is that when the sweep plane is rotated greater than π radians it becomes possible for the same points on an object to intersect both the positive and negative radius sides of the sweep plane.

The second problem is illustrated in Figure 6. Consider the planar face represented by the plane $x = 4$ as shown in Figure 6a. If the face is bounded at height of $z = \pm 2$ and the algorithms presented above are used to sweep from θ_0 to θ_1 the resulting plot in the rh-plane will result as shown in Figure 6b. The reason that the left side of the figure is left open is due to the manner in which the wireframe is swept. As the $z = 2$ edge is swept, the h value of the edge is constant and the r value starts at some value greater than 4 ($= \sqrt{(y_0 * y_0 + 4*4)}$, decreases to 4 at $y = 0$ and then increases again as θ reaches θ_1. When one of the edges at the constant θ boundaries is determined (via clipping and closure) a constant radius edge is generated in the rh-plane. Similar results occur at the $z = -2$ and θ_0 boundaries respectively. For the case when y_0 and y_1 are the same angular distance from the x-axis, the open ended wireframe of Figure 6b will be generated. This is called the minimal radius edge problem and is solved below.

Dealing with Rotational Angles Greater Than π

There are two ways of dealing with rotational angles greater than π. If the first clipping algorithm is used (clipping versus the boundaries separately) then some modifications need to made in the way that edges are clipped by the planes representing the wedge boundaries. Two potential types of modifications would work in this case. The first would be to institute a clip against a half-plane using the origin as the boundary of the half-plane and then clipping against the two half-planes which represent the wedge boundaries. The second technique would involve clipping against the two full-planes representing the extended wedge boundaries and performing a union of the resulting edges. Neither of these options is very desirable, in the first clipping against a half-plane is not well defined and in the second the union operation can be computationally expensive.

When simultaneous clipping against both the wedge boundaries is used, the options for solving this problem are not straightforward. The added difficulty here is that it is now possible for an edge to have both end-points in the unclipped zone and yet still have a portion of itself in the clipped zone. When this is the case, two possible configurations are possible. The first is that the edge should be closed via the radius-equals-zero point; a problem that is already solved. The second configuration describes the situation when the clipped wedge cuts the object into two sections thus causing the face under study to become two faces. In this case, each face with this configuration would have to be clipped and closed twice, once for each side.

If it is only important to know the boundaries of the sweep shadow and extra lines internal to the shadow itself are not important or are being removed in a later step, then the following technique can be used for both independent and simultaneous clipping operations. Since it is demonstrated in previous sections how to clip and close when the rotation angle is less than or equal to π, this problem can be divided into sweeps with angles less than or equal to π. The most obvious choice

of divisions is for one sweep with an angle of π and the other with a sweep of θ_{max} minus π. This requires two extra passes of each face (for a total of four) through the clipping and closure routines, but it will determine the full boundary of the sweep shadow. One important consideration is to keep track of whether the clipped edges have a positive or negative radius since the second sweep wedge begins at angle π and not at angle zero.

Solving the Minimal Radius Edge Problem

The problem illustrated in Figure 6 is not as simple to solve as that of rotation angles greater than π. The problem stems from the fact that the wireframe of an object's faces does not necessarily define the radial boundaries of those faces. By inspection it can be seen that the maximum radius of any point on a plane is infinite, therefore, the wireframe boundary of a face on that plane will define the maximum radii of the points on that face. The wireframe, though, does not always specify the minimal radii points on that face and this is the situation depicted in Figure 6. Thus, it is necessary to locate and sweep the minimal radius boundary for each face.

A planar surface represented in cylindrical coordinates is not simply defined. A brief look at the mathematics, though, reveals that the problem is not difficult to solve. Start with the basic equation for a plane.

$$Ax + By + Cz = D$$

By fixing the value of z and solving for the line on the plane defined by this fixed z yields:

$$y = -\frac{A}{B}x + \frac{D - Cz}{B}$$

Use this to compute the radius squared, and take the partial derivative of this square with respect to x and set it equal to zero. This describes the point on that line where the radius is minimized.

$$r^2 = x^2 + y^2 = x^2 + (\frac{A}{B}x + \frac{D - Cz}{B})^2$$

$$\frac{\partial r^2}{\partial x} = 2x(1 + \frac{A^2}{B^2}) - 2\frac{A}{B^2}(D - Cz) = 0$$

$$x = \frac{A}{(A^2 + B^2)}(D - Cz)$$

By allowing z to vary, a line representing the minimum radius edge of the plane is defined. This line can then be treated like any other edge on the face.

There are a few special cases for the proceeding calculations. For the case when $C = 0$, the values for x and y are constant throughout the plane. When both $A = B = 0$ this is the case where $z = D$ and the plane has a minimal radius of zero which is captured by the closure algorithm and can be ignored. For the final special case when $B = 0$, the normal of the plane of the face has no y component and the value of zero for y can be used.

Now that the minimum radius line for the plane representing the face is defined some further processing needs to be performed. First, it must be determined whether the line even intersects the area of the face. This is just the Polygon Intersected Edges problem as defined by Preparata and Shamos [Prep, p. 313]. If the line does intersect the polygon defining the face then extra edges

must be added where appropriate. (This is one edge for a convex polygon and possibly more for a non-convex polygon.)

A solution to this problem is to extend the single-shot polygon inclusion test so that all the desired edges are obtained [Prep, pp. 41-43]. The basic idea of the single-shot polygon inclusion test is to determine whether a point is located inside a polygon by drawing a ray from that point to infinity and counting the number of intersections with the edges of the polygon. An odd number of intersections means the point is inside and an even number means the point is outside. The minimum radius line is used to define the direction of this ray. Rays are drawn in both directions away from the point and by sorting all the intersection points, a set of minimal radius edges can be determined by alternately labeling each intersection point with the rays as inside or outside the polygon defining the face.

There are a few non-trivial problems associated with this approach which are only briefly mentioned by Preparata and Shamos. These involve degenerate intersections between the line and the edges of the polygon. The first arises when the line intersects the polygon at a vertex. Not only does this intersection point intersect two edges, it is also quite possible that the line is only grazing the polygon and does not enter or leave it at that point. Whether this type of intersection represents a true intersection with the polygon can be determined by looking at the sign of the sine of the angle between the line and the edges of the polygon when both of the edges are directed in the same direction about the polygon. This information is readily determined by using the vector cross product.

The second problem arises when the line and an edge overlap each other. This, though, is just an extension of the vertex intersection problem. By looking at the edges on both ends of the overlapping edge it can be determined whether this type of intersection defines a crossing point or not.

The Complete Solution

In Figure 2, a simplified algorithm is provided which attempts to solve the rotational sweep shadow problem. In previous sections, this algorithm is shown to be mostly correct, but lacks certain features which leave the problem incompletely solved. Figure 7 depicts an enhanced version of the algorithm in Figure 2 with added steps for solving the problems associated with large sweep angles and minimal radius edges.

Even though the solution to this problem requires up to four passes through the clipping and closure routines for each wedge, the time complexity is still linear with respect to the number of edges in the obstacles. The only portion of the solution which is not linear is the sorting of intersection points found in the minimal radius edge solution. In the worst case this sorting can be done in nlogn time, but it is more likely that only zero, two, or four points need be sorted for a face with most faces possessing zero or two intersections for reasonable objects.

Results

A rotational sweep shadow generator using the techniques of simultaneous clipping as described above has been implemented in C on a Sun-3 workstation. It is incorporated within a robotic assembly planning system and is used to analyze mating trajectories, grasp sites, and grasp types for the case of simple rotations. Figure 8 shows the results of this algorithm as it encounters a cube. In both the cases, the extra internal lines of the shadow are left to aid in visualization and because their presence does not affect the operation and results of the larger system.

Figure 8a shows the cube when the center of the rotation lies within the cube. The rotation angle is 2 radians and the apparent extra set of lines internal to the right and left sides of the figure result

from the minimal radius edge calculations. Figure 8b shows then same cube when the center of the rotation lies outside of the cube. The triangular patch to the right shows the extra face caused by clipping and closure at one of the wedge boundaries. The two vertical lines which are tangent to two of the curved edges are each minimal radius edges. In the case of the right-most one, its presence in necessary.

Conclusion

It has been shown that the rotational sweep shadow problem yields theoretical as well as practical results. Two approaches to clipping at the wedge boundaries are presented. The approach using simultaneous clipping is implemented since it shows a slight computational superiority. This is due to the fewer number of extra clips and closures performed because all the clipping is performed in one pass through the edge data rather than two. This makes for a reasonable trade-off against the higher computational intensity of computing the midpoint and the double edged closure through $r = 0$.

The test results of the previous section demonstrate that the effects of the minimal radius edge are important if the entire sweep shadow is to be determined. The results generated from the algorithm are also useful in solving robotics problems associated with assembly sequence analysis, trajectory planning and analysis, and the analysis of grasping options. The further advantage offered by the use of constant cross-sectional area motions is the reduction of the analysis of gripper envelopes to two-dimensional areas as opposed to three-dimensional volumes.

Future Research

There are a number of ways in which this research can be enhanced. The addition of more complex surface types such as cylinders and spheres can be considered. It would also be possible to use the insights gained in solving the rotational sweep shadow in solving the other classes of constant cross-sectional area motions (e.g., helical paths and constant radius curves). Further applications to this type of algorithm might also be found in areas other than robotics.

Acknowledgement

This work was supported in part through NASA grant NAGW-1333.

BIBLIOGRAPHY

[Blinn] Blinn, J. F., "Optimal Tubes," *IEEE Computer Graphics and Applications*, September 1989, pp. 8-13.

[De Fazio87] De Fazio, T. L. and Whitney, D. E., "Simplified Generation of all Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, December, 1987, pp. 640-658.

[De Fazio88] De Fazio, T. L. and Whitney, D. E., "Correction to 'Simplified Generation of all Mechanical Assembly Sequences'," *IEEE Journal of Robotics and Automation*, December, 1988, pp. 705-708.

[Foley] Foley, T. A., et al., "Visualizing Functions Over a Sphere," *IEEE Computer Graphics and Applications*, January, 1990, pp. 32-41.

[Hearn] Hearn, D. S. and Baker, M. P., *Computer Graphics*, Prentice Hall, Englewood Cliffs, NJ, 1986.

[Hoffman] Hoffman, K. and Kunze, R., *Linear Algebra*, Prentice Hall, Englewood Cliffs, NJ, 1971.

[Homem86] Homem de Mello, L. S. and Sanderson, A. C., "AND/OR Graph Representation of Assembly Plans," *AAAI-86 Proceedings of the Fifth National Conference on Artificial Intelligence*, 1986, pp. 1113-1119.

[Homem88] Homem de Mello, L. S. and Sanderson, A. C., "Task Sequence Planning for Assembly," *IMACS World Congress '88 on Scientific Computation*, July 1988.

[Ko] Ko, H. and Lee, K., "Automated Assembling Procedure Generation from Mating Conditions," *Computer Aided Design*, January-February, 1987, pp. 3-10.

[Mirolo] Mirolo, C. and Pagello, E., "A Solid Modeling System for Robot Action Planning," *IEEE Computer Graphics and Applications*, January, 1989, pp. 55-69.

[Prep] Preparata, F. P. and Shamos, M. I., *Computational Geometry an Introduction*, Springer-Verlag, New York, NY, 1985.

[Selby] Selby, S. M., ed., *Standard Mathematical Tables*, The Chemical Rubber Co., Cleveland, OH, 19th. Ed., 1971.

[Turner] Turner, J. E., "Accurate Solid Modeling Using Polyhedral Approximations," *IEEE Computer Graphics and Applications*, May, 1988, pp. 14-28.

[Wang] Wang, W. P. and Wang, K. K., "Geometric Modeling for Swept Volume of Moving Solids," *IEEE Computer Graphics and Applications*, December, 1986, pp. 8-17.

[Welch] Welch, Henry L., *Robot Independent Assembly Sequence Planning*, PhD. Thesis, Rensselaer Polytechnic Institute, August 1990.

Captions of Figures

Figure 1. Basic structure chart for an assembly sequence planner.

Figure 2. A simplified algorithm for solving the rotational sweep shadow problem.

Figure 3. An arbitrary cylindrical coordinate system defined by a spatial rotation.

Figure 4. The possible clipping arrangements for a straight line edge intersecting a cylindrical wedge.

Figure 5. The results of clipping and closing a polygonal face against a cylindrical wedge. The heaviest lines represent the final shape of the face.

Figure 6. An illustration of the minimal radius edge problem.

Figure 7. A complete algorithm for solving the rotational sweep shadow problem.

Figure 8. The rotational sweep shadow of a cube. (a) With the center of rotation inside the cube. (b) With the center of rotation outside of the cube.

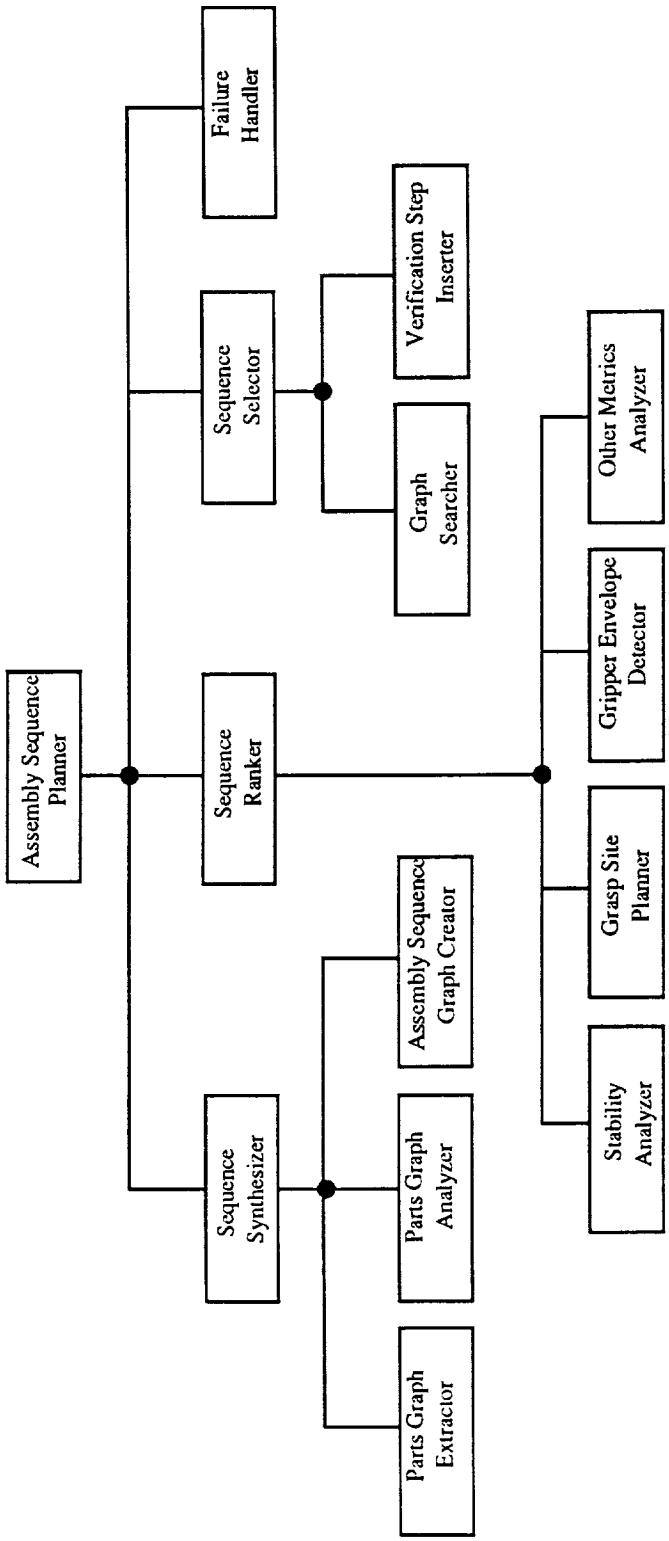
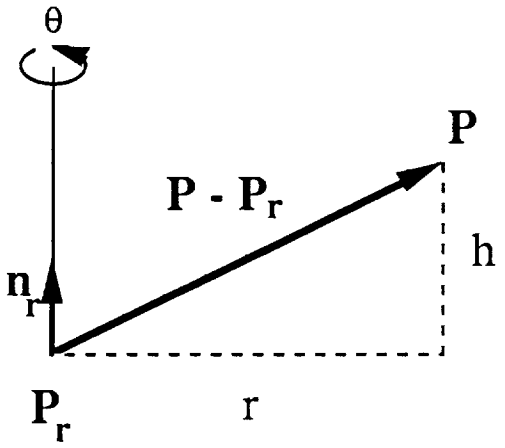


Figure 1 - Basic structure chart for an assembly sequence planner

```
procedure GENERATE_ROTATIONAL_SWEEP_SHADOW;
    specify and define the cylindrical coordinate system for the sweep;
    for each object in the environment;
        for each face on the object;
            for each edge on the face;
                convert the edge to the sweep coordinate system;
                sweep the edge and clip as necessary;
                close with the previous edge if necessary;
            close the face between the initial and final vertices;
end-procedure;
```

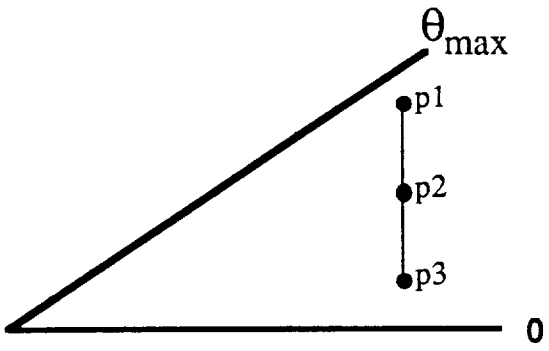
Figure 2. A simplified algorithm for solving the rotational sweep shadow problem.



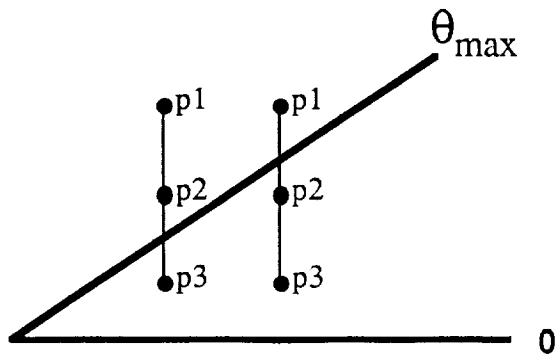
$$h = (P - P_r) \cdot n_r$$

$$r = \sqrt{(|P - P_r|^2 - h)^2}$$

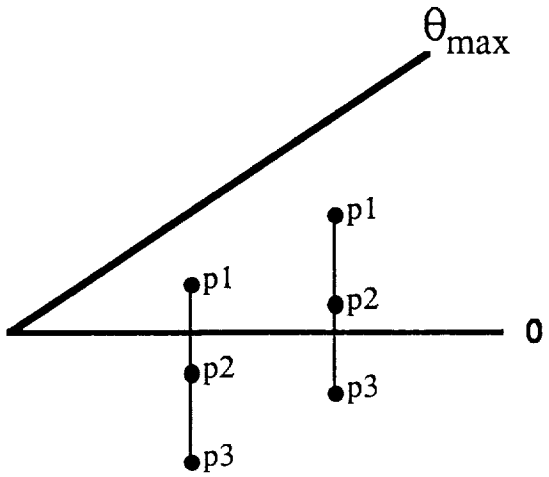
Figure 3. An arbitrary cylindrical coordinate system defined by a spatial rotation.



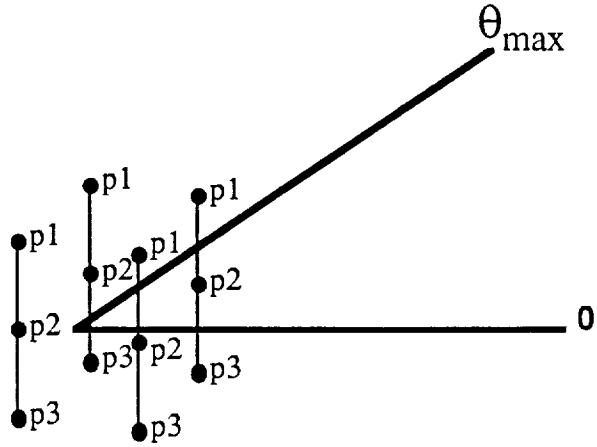
(a) Use entire edge



(b) Clip p1 end to θmax



(c) Clip p3 end to θ = 0



(d) if θ1 > θ2 and θ2 > θ3 ignore
otherwise clip p1 end to θmax
clip p3 end to θ = 0

Figure 4. The possible clipping arrangements for a straight line edge intersecting a cylindrical wedge.

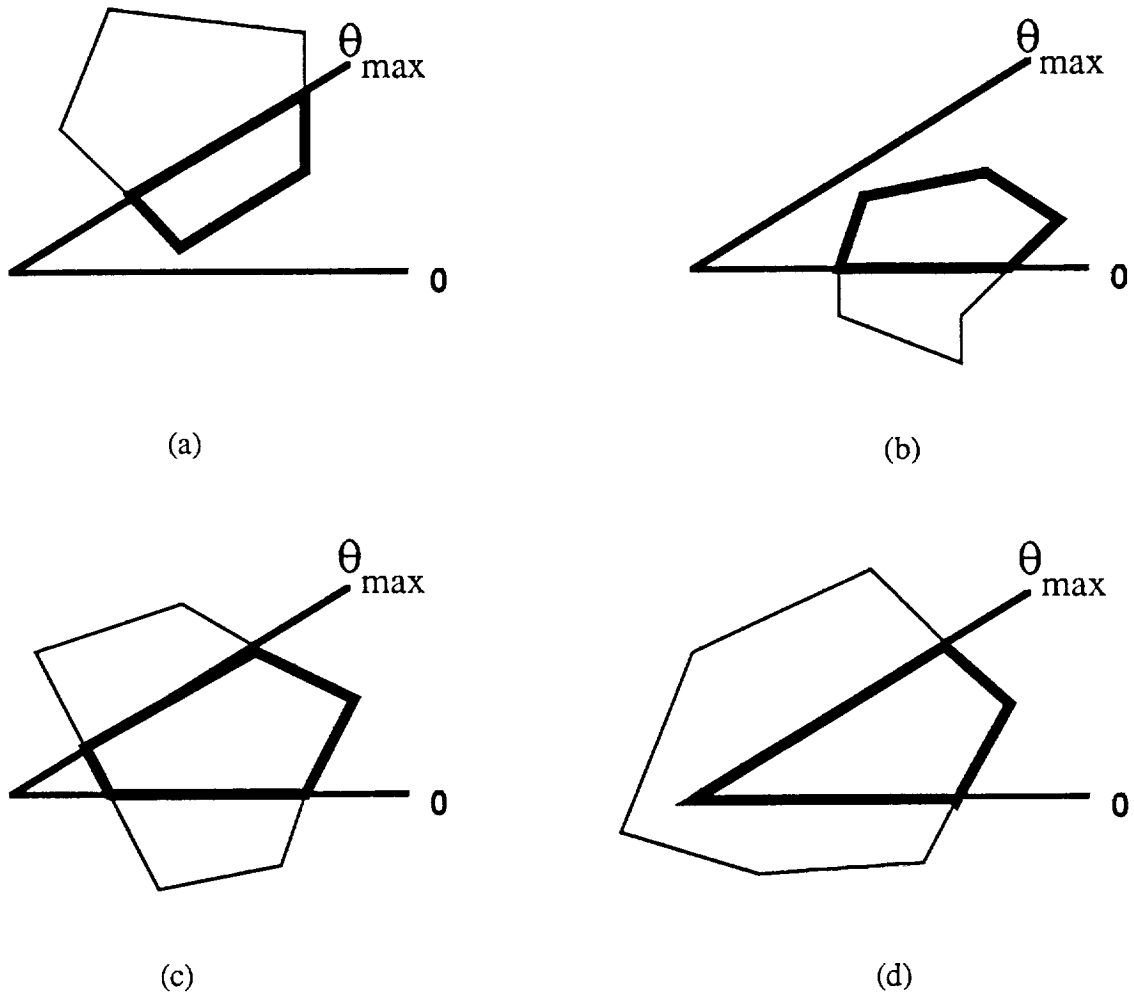
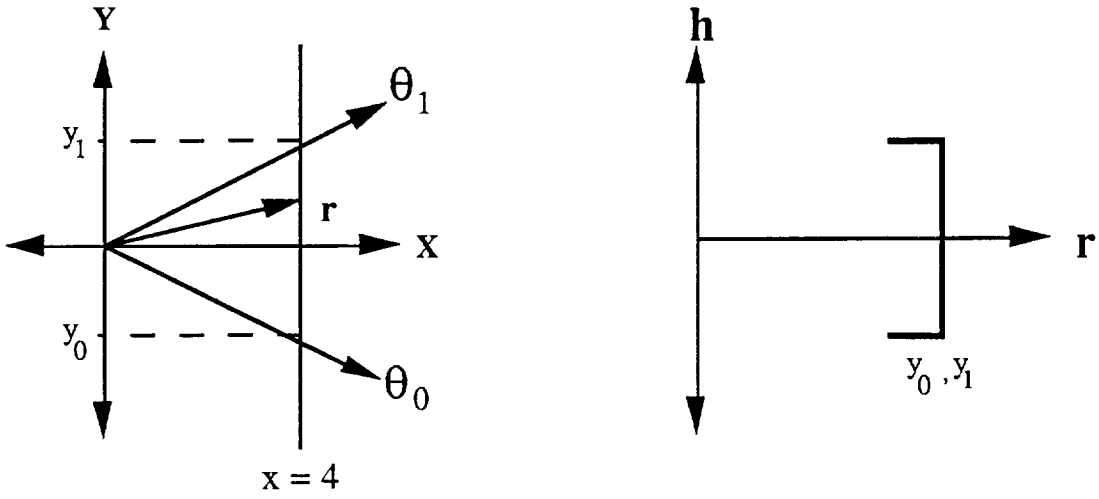


Figure 5. The results of clipping and closing a polygonal face against a cylindrical wedge. The heaviest lines represent the final shape of the face.



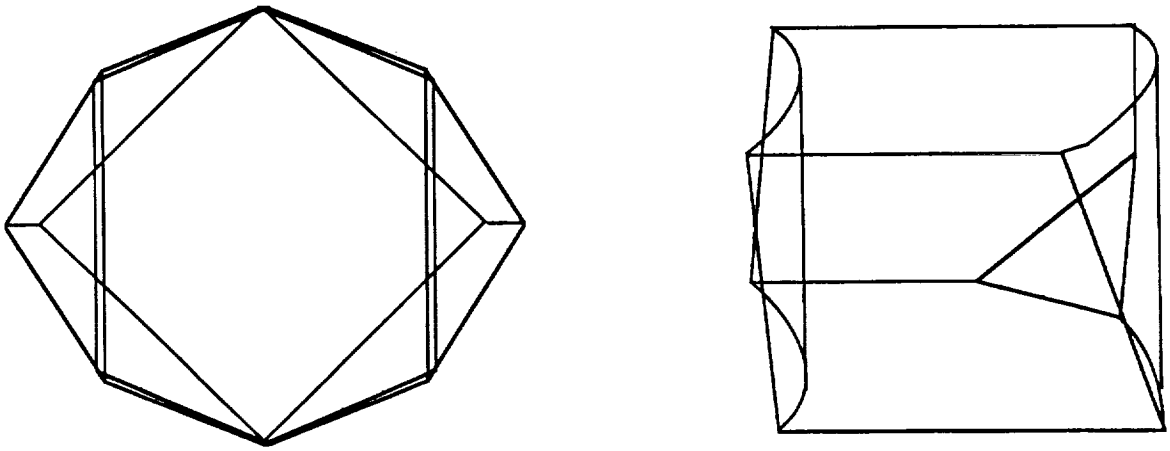
(a) A planar face at x = 4

(b) The results of clipping and closing the wireframe of part (a).

Figure 6. An illustration of the minimal radius edge problem.

```
procedure GENERATE_COMPLETE_ROTATIONAL_SWEEP_SHADOW;
    specify and define the cylindrical coordinate system for the sweep;
    for each object in the environment;
        for each face in the environment;
            find the height of the face at r=0;
            find the minimal radius edges;
            for each minimal radius edge;
                if θmax > π then;
                    sweep in two parts;
                else;
                    sweep in one part;
                end-if;
            for each edge on the face;
                convert the edge to the sweep coordinate system;
            if θmax > π then;
                for each edge on the face;
                    sweep for θ < π and clip as necessary;
                    close with the previous edge if necessary;
                close the face between the initial and final vertices;
                reduce θmax by π;
                rotate all the data π radians;
            end-if;
            for each edge on the face;
                sweep the edge and clip as necessary;
                close with the previous edge if necessary;
            close the face between the initial and final vertices;
end-procedure;
```

Figure 7. A complete algorithm for solving the rotational sweep shadow problem.



(a) (b)

Figure 8. The rotational sweep shadow of a cube. (a) With the center of rotation inside the cube. (b) With the center of rotation outside of the cube.